Amendments to the Specification

Please amend the paragraph beginning on page 1, line 5, as follows:

This application is related to U.S. Patent Application No. ~~_____~~ 10/645,729 (Attorney Docket No. ADAPP244), filed on the same day as the instant application and entitled "METHOD AND APPARATUS FOR ACCELERATING TEST CASE DEVELOPMENT." This application is hereby incorporated by reference in its entirety.

Please insert this paragraph after page 6 line 18, as follows:

Figure 6 is a block diagram illustrating the execution flow for a client server system configured to accelerate test case development in accordance with one embodiment of the invention.

Please insert these paragraphs before the paragraph beginning on page 13, line 1, as follows:

Figure 6 is a block diagram illustrating the execution flow for a client server system configured to accelerate test case development in accordance with one embodiment of the invention. The client server system includes two main components, i.e., server component 222 and client component 224. Client component 224 executes client script 200. In one embodiment, client component 224 is configured to read a text file. In block 202 the client script, which was initiated in block 200, submits the test case. The test case may include a group of tasks, i.e., macro tasks. The tasks associated with the test case is written as a text file. It should be appreciated that since the initialization sequence is stored on the server side, the initialization sequence is not needed as part of the test case being submitted to server component 222. The client script then creates a testinput.txt file which is used to

submit to server component 222 and the client script then waits for the result of the server execution as illustrated in block 206.

The testinput.txt file of block 204 of Figure 6 is a pre-processed file in which syntax and format errors associated with the testcase.txt file have been checked. Thus, a verified file is generated that is then moved to the data stream where the server component 222 is running. In one embodiment, testinput.txt of block 204 includes text-based tasks associated with precompiled hardware description language based tasks. Server component 222 maintains the server in an initialized state as mentioned above. Here, chip initialization text of block 210 is executed by a simulation server. The simulation server executes the re-set and chip initialization sequence in block 212. The initialized server is maintained in the initialized state, awaiting for a test case to be delivered from client side 224 as illustrated in block 214. Thus, when the testinput.txt file block 204 is moved into the data stream of the server component 222, the simulation server is available to immediately process and execute the test case associated with the testinput.txt file. Accordingly, the simulation server executes the test and publishes the test case results in block 216.

Once the test has completed, the simulation server is uninitialized in block 218 of figure 6. The chip initialization test is then re-run in order to initialize the simulation server and the process is repeated as illustrated on the server component 222 of Figure 6. Upon completion of the test case in block 216, the server moves the test case results to the client side as a result.log in block 220. The result.log of block 220 is then communicated to the client script which is waiting in block 206. Thus, the system described above eliminates the need for waiting for any database to be linked or the need for re-setting the chip and completing the initialization sequence prior to executing the test case